Development of a Pre- and Post- Processing Environment for ACES that Incorporates JView Software for Visualization

Project Summary Presentation

AATT RTO 80

Deliverable 5

September 14, 2004

Jesse Aronson, SAIC / jaronson@sito.saic.com / 703-907-2553
Michael Gallan, SAIC / mgallan@sito.saic.com / 703-907-2507

Agenda

- Task Goals
- Task Overview
- JView Evaluation Results
- Task Products
  - SPADES Database
  - Weather Editor / FDS Viewer
  - ASRTV Viewer

AATT RTO80 Goals

- **Separate the ACES Simulation Control and VST functions**:
  - Within ACES, the Simulation Control and VST functions are not cleanly separated. Simulation Control and visualization are combined in a single application.
- **Define the pre-processing environment**:
  - Create a concept for a pre-processing environment for ACES.
- **Prototype pre-processing software**:
  - Create prototypes to demonstrate the pre-processing system concept.
- **Demonstrate Visualization of ACES Data Using JView**:
  - Investigate the applicability of the JView 3D API in visualizing ACES runtime and non-runtime data.

AATT RTO80 Approach:

Scenario Processing and Data Environment for Simulations (SPADES)

- **Separate the ACES Simulation Control and VST functions**:
  - Create a separate JView-based visualization federate - ASRTV
  - Leave all control functions in VST
  - Make use of ability to disable visualization in VST
- **Define the pre-processing environment**:
  - Created a concept based on a database for ACES data and a suite of tools to create, visualize and manipulate these data
  - Design database schema for ACES data
- **Prototype pre-processing software**:
  - Prototype database implementation of schema
  - Weather editor, FDS visualization prototype
  - Demonstrate connectivity to ACES with weather data as an example
- **Demonstrate Visualization of ACES Data Using JView:**
  - JView used for graphics in both ASRTV runtime viewer and Weather Editor / FDS Viewer

All products are usable with ACES: not "throw-aways"

Project Products / Major Deliverables

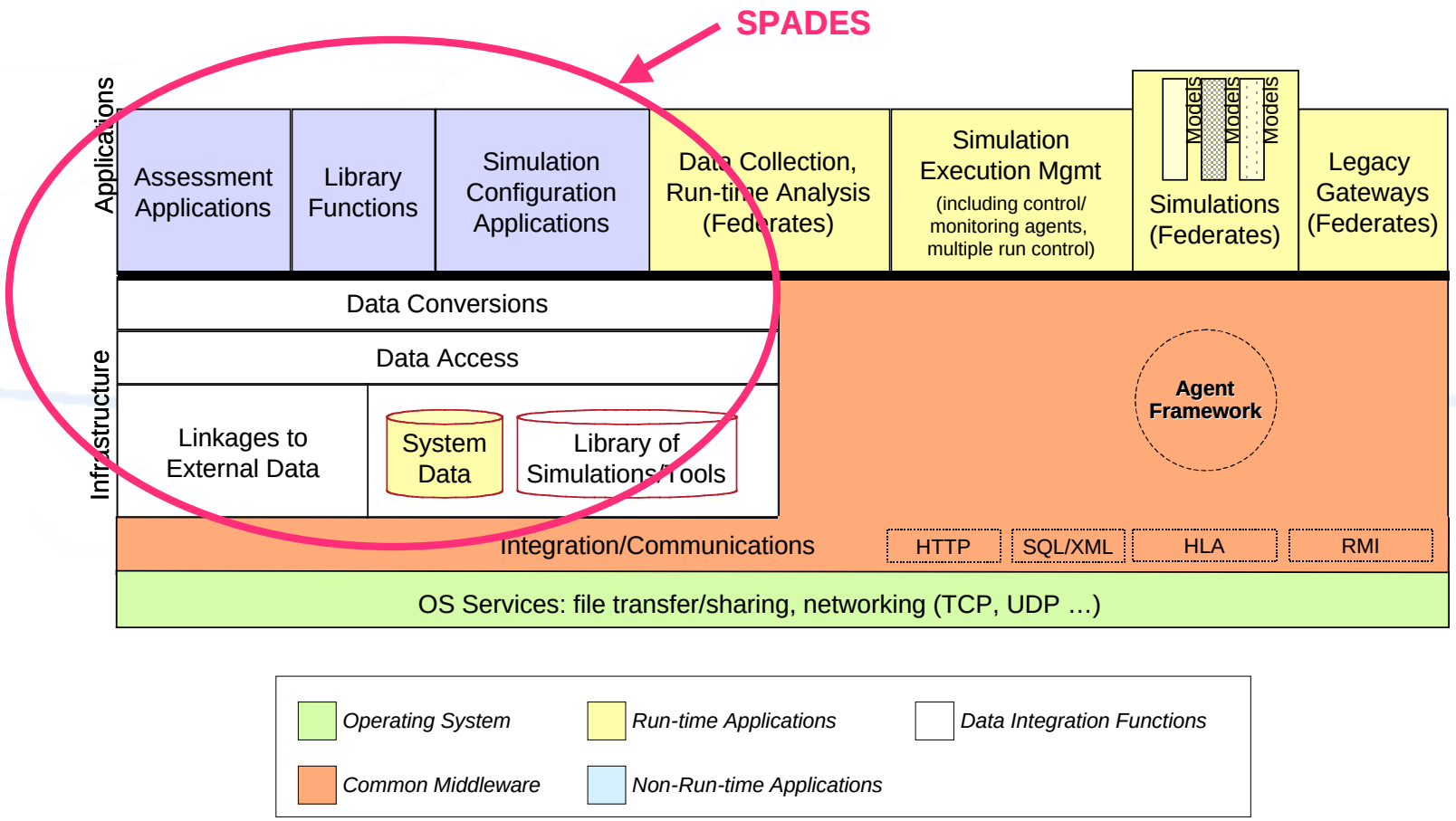
- SPADES Database Schema
  - Schema Design Document
  - Prototype MySQL Implementation
- ACES-SPADES Runtime Viewer (ASRTV)
  - Engineering, Software Design Documents
  - User Guide
  - Build 2.0.3 and Build 3.0.1 – compatible software
- Weather Editor / FDS Viewer
  - Engineering, Software Design Documents
  - User Guide
  - Build 3.0.1 – compatible software
- Final Report

Process and Technologies

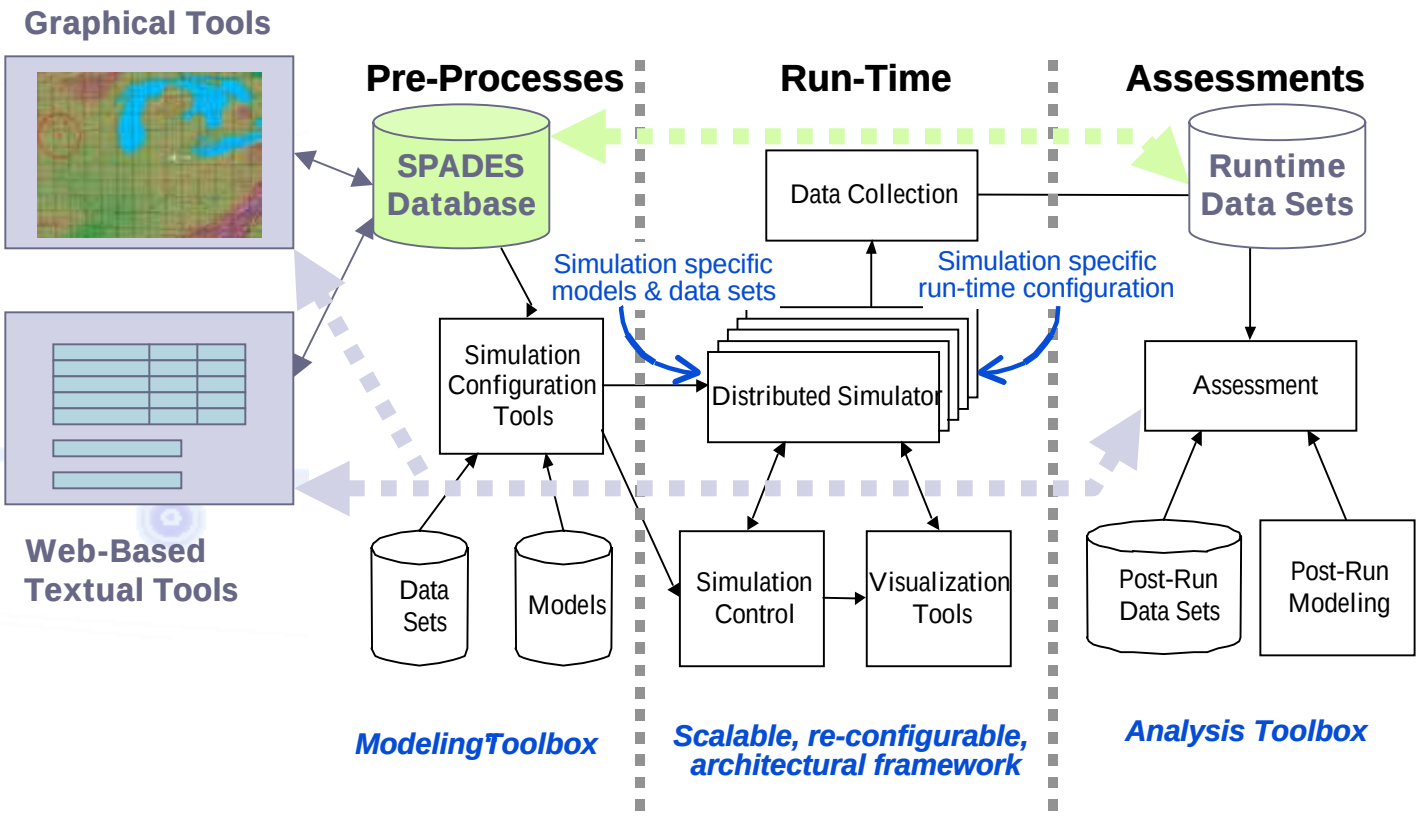
- Platform independent software
  - Java, JSP, JView
  - MySQL with JDBC
- Adhere to ACES coding standards
- Use NASA Bugzilla Server
  - Defect Tracking
  - Suggestions for Enhancements
- Development CM at SAIC
- Use SAIC RPM Tool for Requirements Management

SPADES Concept Overview

ACES Architecture (from ACES SSDD)



SPADES Concept



JView Evaluation

JView Strengths

- Allows users to develop 3D applications without prior knowledge of OpenGL
- Geared towards simulations
  - Coordinate systems
  - Terrain
  - Dynamic graphics
- Developers are focused on optimizing performance
- Developers are willing to accommodate questions
- Overall class design is logical and uncluttered, especially when compared to Sun's Java3D API
- The price is right

JView Issues

- Documentation
  - "Introduction to JView" document was incomplete.
  - No real programmers guide
    - Fairly extensive set of example programs
  - Many JavaDoc comments were missing or assumed a familiarity with OpenGL
  - Broken or unimplemented features were not noted
- Feature Set
  - 2D API, "lenses" not working
  - CLOD terrain functioned poorly in v1.1, said to be much better in v1.2.

JView Issues (cont.)

- Support is directly via developers, but
  - Developers often difficult to reach by phone or e-mail
- Loose release process:
  - Classes and properties were removed or altered between releases without notice
  - Unclear whether bug fixes were committed to the main code repository
  - Testing is clearly not extensive
  - No public bug database
- Small user community

Alternatives to JView (p. 1 of 2)

- Low-level graphics APIs: JOGL, GL4Java, LWJGL
  - Offer complete control over graphics and performance.
  - Require OpenGL expertise and greater development time.
- GIS-specific libraries: OpenMap, MapObjects, ILOG JViews
  - Generally focus on 2D display with little 3D support.
  - Emphasis on web/database connectivity.

Requirements for cross-platform operation, map or terrain, Java API are constraining

Alternatives to JView (p.2 of 2)

- Scene graph-based Java APIs:
  - Java3D (https://java3d.dev.java.net)
    - Pros: Well documented, source code recently opened by Sun for community development.
    - Cons: Reputation for poor performance, no recent releases, limited access to low-level OpenGL calls.
  - Xith3D (http://www.xith.org)
    - Pros: Open source, allows direct OpenGL calls, emphasis on performance, supports JOGL and LWJGL.
    - Cons: Limited documentation, not thread-safe.
  - jME (http://www.mojomonkeycoding.com/)
    - Pros: Well documented, open source, active community.
    - Cons: Currently only supports LWJGL.

JView is a scene graph API

SPADES Database Overview

Database Schema Design Goals

- Capture ACES Input Data in a more centralized way
- Introduce unifying thread to tie together disparate ACES data elements
  - e.g., cto7sim data vs. simstartup data vs. scenario data
- Support for multiple researchers, concept developers, studies, runs
  - Archiving and Sharing
- Provide a Foundation for Pre/Post-processing Tools
- Develop proof of concept implementation in MySQL
  - Foundation for other RTO80 prototyping tasks

Categories of Data

- Static Lookup tables
  - Static data lists (ACES static data)
- Configuration Data
  - Describes how an ACES run will be executed on a set of computers (ACES configuration data)
- Experiment-Specific Data
  - Items that might commonly vary from run to run
  - Scenario data, overrides for many default parameter values
- Reusable Experiment Data
  - Data that is variable but which might get used multiple times unchanged across runs (e.g., a Flight Data Set File)
- Data Collection Data
  - Local data collection configuration
- Editor Tool Data
  - Data that is specific to SPADES tools

New Concept: Experiment

- SPADES introduces the notion of an *Experiment*
- Ties together the six data categories on the previous slide
- Associates a set of experiment data with a user
- Could be used for Batch Run environments

New Concept: Weather

- Explicit representation of convective weather
  - "Weather System" made up of multiple "Weather Affected Areas" (WAAs)
  - Each WAA contains:
    - Center
    - Radius
    - Min, max elevations
    - Times of applicability
    - Impact Parameters: airport state, sector capacity % multiplier
- Used in SPADES Weather Editor Tool
- Weather generation algorithm generates ACES scenario events based on this data
  - Weather editor generates this data

ACES-SPADES
Weather Editor / FDS Viewer
Overview

Task Goals

- Develop a concept for an ACES data pre-processing environment
- Demonstrate the concept through a prototype
- Demonstrate the use of JView in visualizing NAS data

Prototype example selected is a Weather Editor

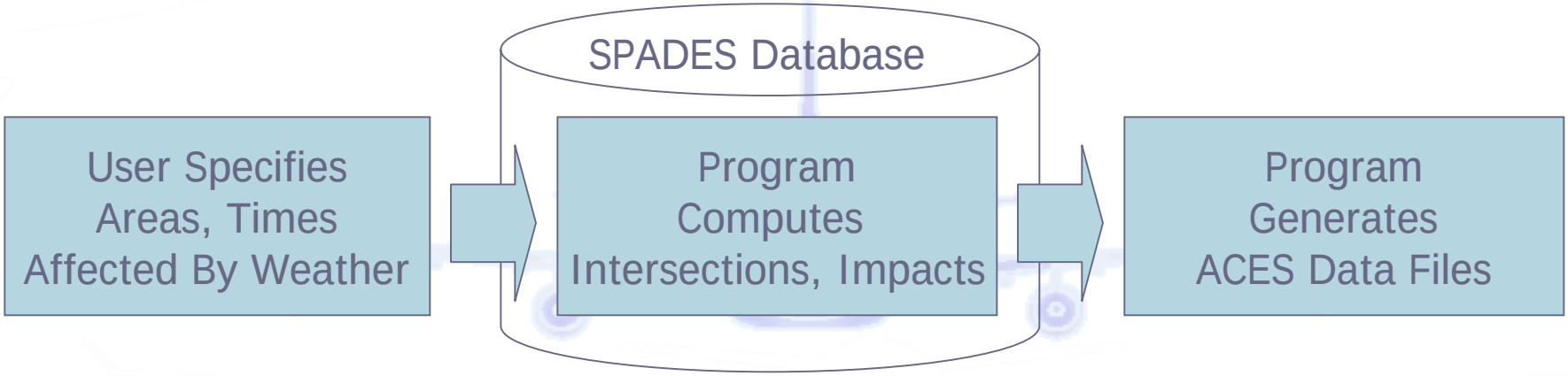
Task Products

- Weather Editor
- Documentation
  - Engineering Design Document
  - Software Design Document
  - User Guide
- Example Weather Editor graphics
  - Demonstrates the use of JView

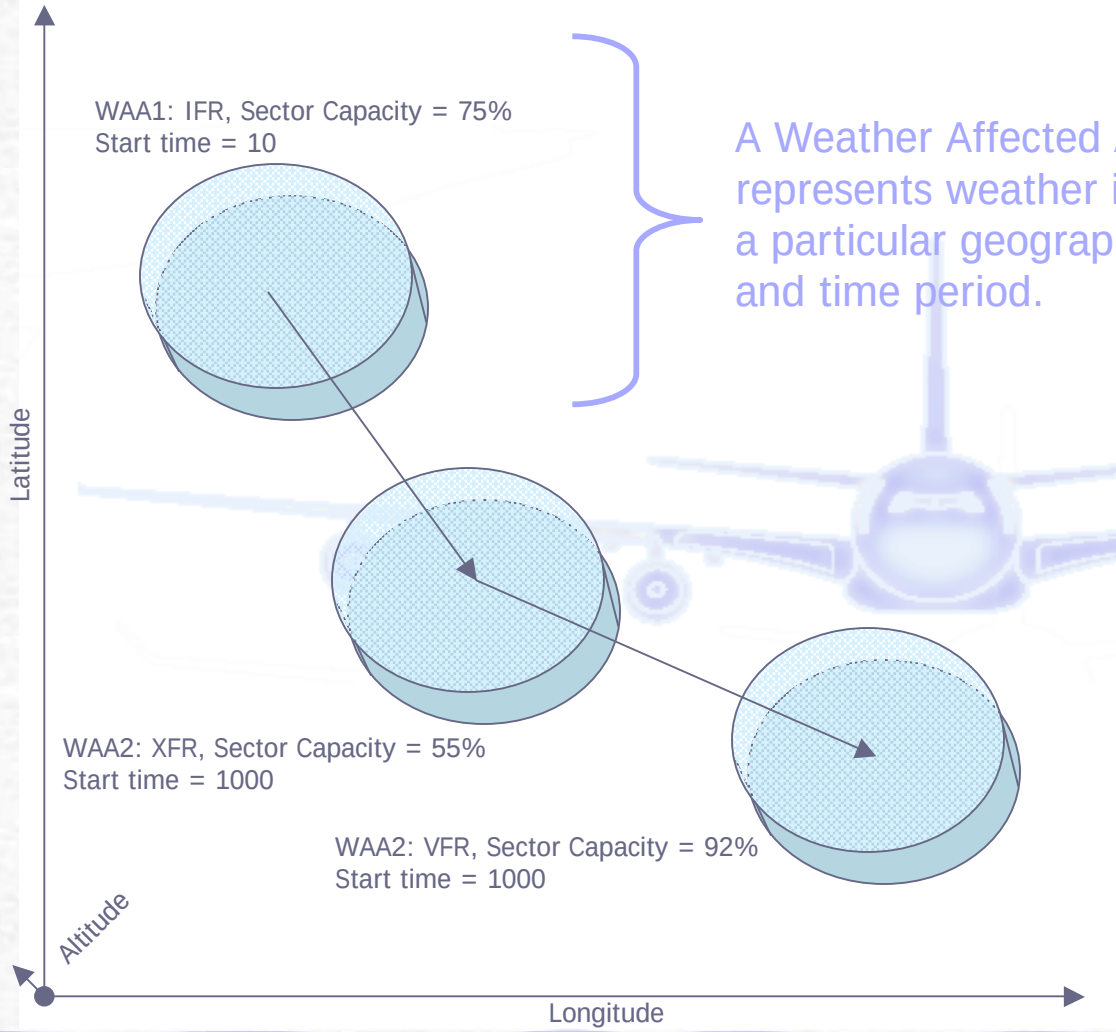
Weather Editor Concept and Highlights

- SPADES Weather Editor implements a graphical “point and click” interface for creating convective weather data for ACES
- Without tools, modeling convective weather in ACES is time-consuming and error prone
  - Weather is represented implicitly through its impacts on individual airports and sectors
  - A moving storm requires manual entry of large numbers of scenario events.
- The editor can also visualize ACES Flight Data Set data as an overlay
- Based on SPADES Database

Weather Editor Process



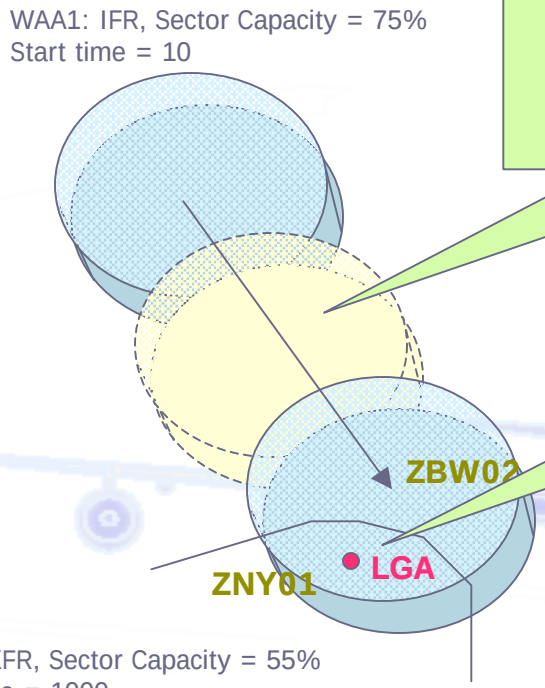
Weather Editor Data Representation



A Weather Affected Area (WAA) represents weather impacts at a particular geographic area and time period.

A Weather System (WS) is a time-ordered collection of WAAs.

Weather Generation Algorithm



Step 1: Interpolate between WAAs as needed to get an uninterrupted time series

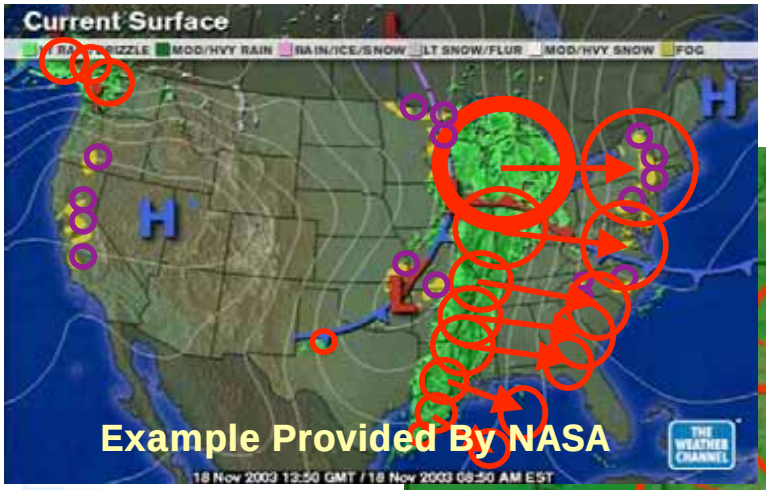
Step 2: Compute intersections between WAAs and airports/sectors

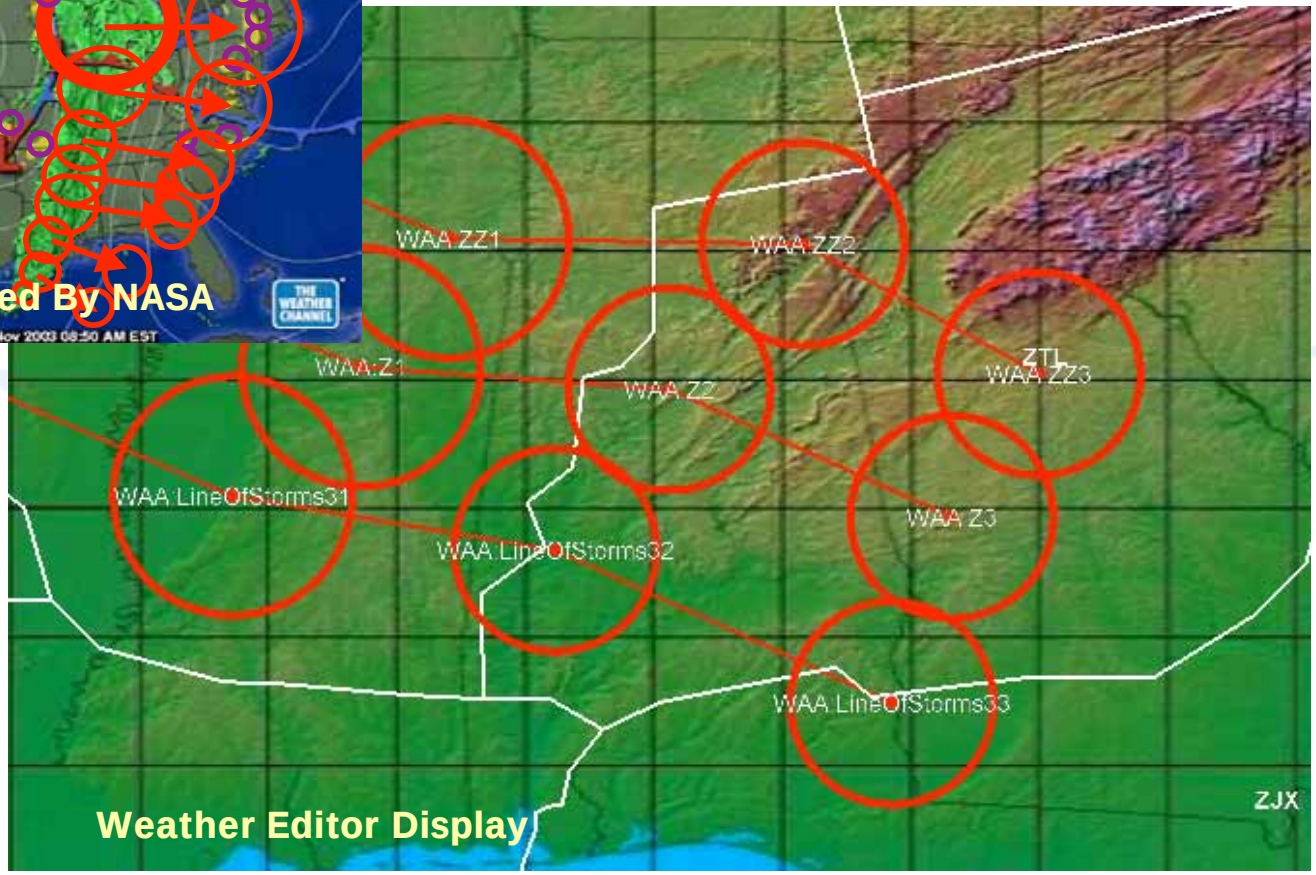
Step 3: Generate corresponding ACES scenario events

- LGA: XFR starting at time 1000
- ZBW02: Capacity = 7 starting at time 1000
- ZNY01 Capacity = 8 starting at time 1000

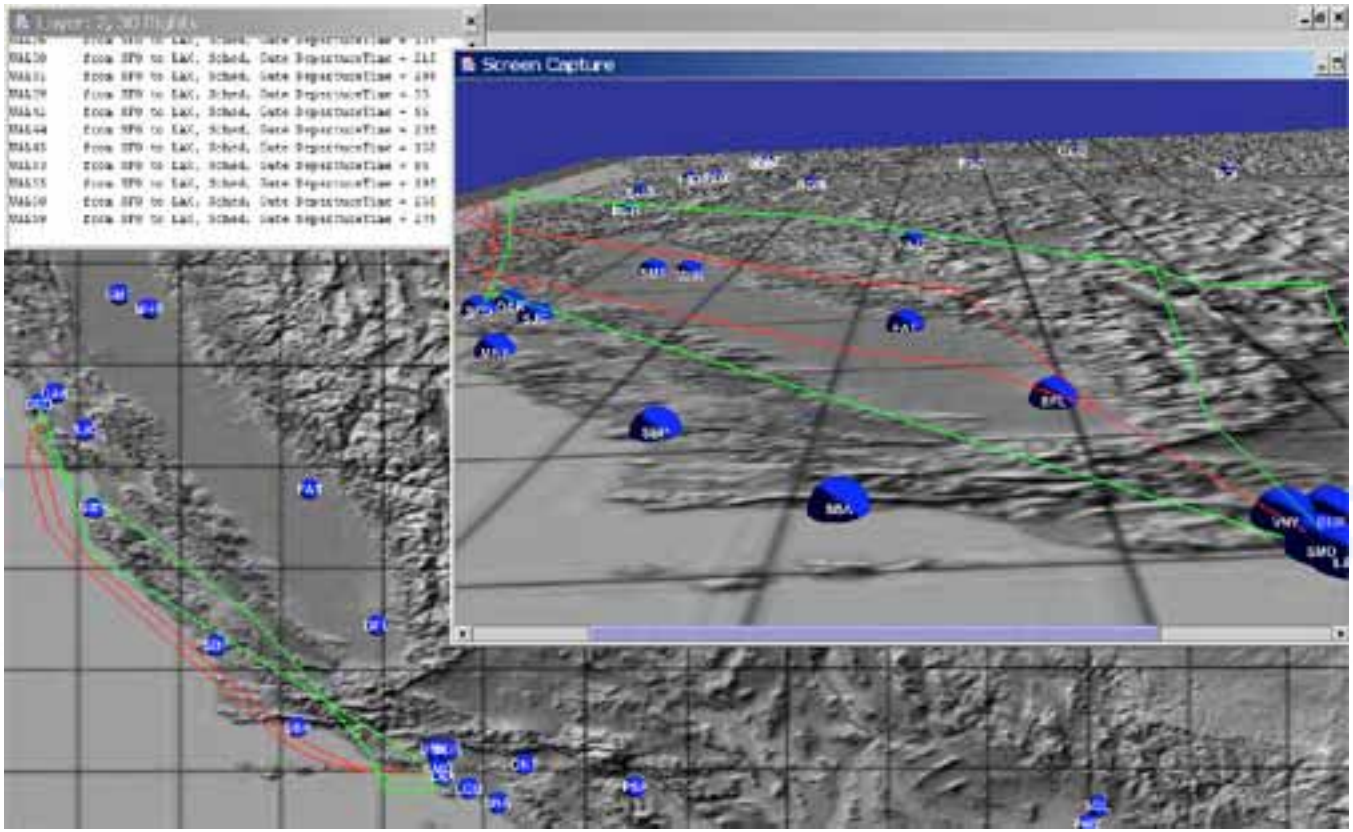
…

Screenshot: A Moving Line of Storms

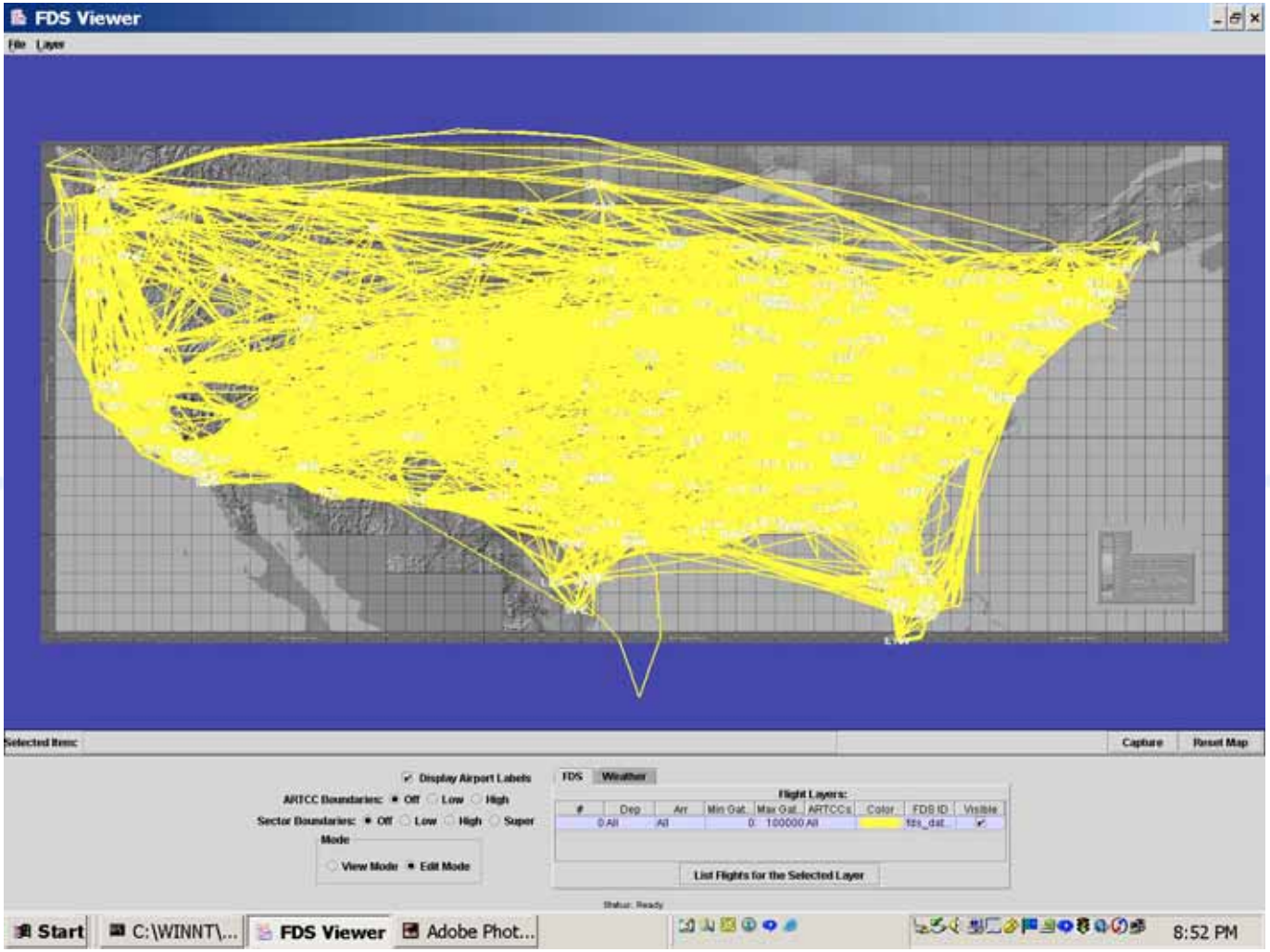




Screenshot: Flight Data Set Visualization



Visualization of 5/17 FDS File



ASRTV Runtime Viewer

Task Goals

- Separate Visualization and Control portions of ACES VST
- Demonstrate the use of JView in visualizing NAS data

From the NASA RTO-80 Task statement:

- " ... us[e] JView libraries to provide a better plan view map with more information available than in the current ACES visualization tool"

ASRTV Highlights

- Adds new visualization functionality
  - Display of flight events
    - Conflicts, boundary crossings, maneuvers, TOC, TOD
  - Aging of trails
  - 3D visualization
  - Full point and click interface
  - Ability to run multiple simultaneous ASRTV viewers
- Maintains all VST visualization displays
- Can run in tandem with VST map if both views are desired
  - VST still used for control functions, but its map can be shut off if ASRTV is being used as the viewer
- Integrated with ACES Build 2.0.3, Build 3.0.1

Task Products

- ASRTV Federate
  - A separate federate that listens to ACES simulation data and provides a 3D visualization
- Modified VST, Simstartup and Batch Files
  - Fully integrates ASRTV with ACES Build 2.0.3.
  - Runs with Build 3.0.1 using Simstratup or batch files
- Documentation
  - Engineering Design Document
  - Software Design Document
  - User Guide
- Example ASRTV graphics
  - Demonstrates the use of JView

Example Uses

- Evaluating complex algorithms involving 2D and 3D maneuvering
- Creation of presentation graphics
- Multiple Views
- Basis for playback
  - Since it is a separate federate not tied to simulation evolution, a playback federate could play data back from LDC through the RTI in fast time and visualize it via ASRTV
- Basis for other visualization
  - Graphics is confined to one agent – could use the underlying federate software as a basis for other graphical/ textual/ analytical visualization

Example

ARTCC

Airport

Flight

Maneuver Event

Conflict Events

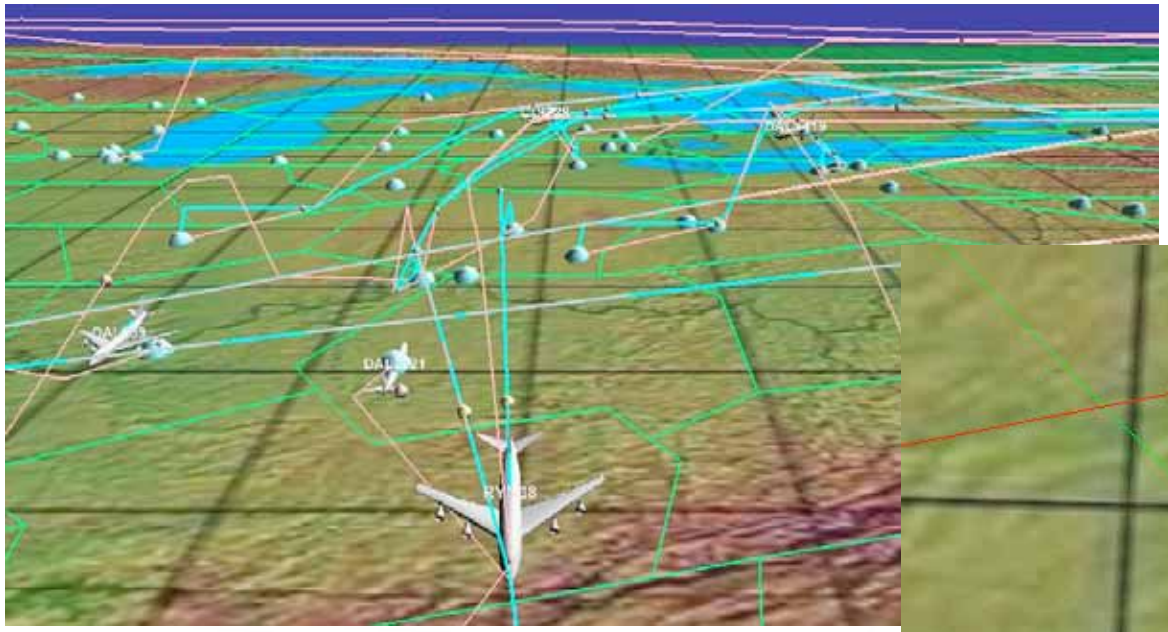
Sector

Flight Plan (pink)

Trajectory showing maneuver (blue)

All map items shown are clickable for more detail

Examples with Airplane Icons



Summary

- Evaluation of JView Completed
  - Use of JView demonstrated
- SPADES Concept developed and demonstrated
  - Schema Design
  - Various prototypes
- There's always room for improvement!